

## WIX1002 Fundamentals of Programming

### Tutorial 10 Polymorphism

1. Create an abstract class `DiscountPolicy`. It has a single abstract method `computeDiscount` that return the discount for the purchase of a given number of item. The method has two parameters `count` and `itemCost`. Derived a class `BulkDiscount` from `DiscountPolicy`. It has a constructor that has two parameters `minimum` and `discount rate`. It has a method `computeDiscount` that compute the discount base on discount rate if the number of item more than minimum. Otherwise, no discount given. Derived a class `OtherDiscount` that compute the discount base on the table below

N (number of Item)	1 – 2	3 – 5	6 – 8	>8
Discount	0	10%	20%	30%

Derived a class `combinedDiscount` from `DiscountPolicy`. It has two parameters of type `DiscountPolicy`. It has a method `computeDiscount` that return the maximum value returned by the `computeDiscount` for the two discount policies. Create a `Tester` class to test the program.

2. Create an interface `Interest` that has a single method `computeInterest` that return the monthly interest based on the balance in the account. Create the `SavingAccount` that implement the interface, the class has an instance variable called `balance`. Define the method to compute interest. The interest rate for saving account is 0.5% per year. Create the `FixedAccount` that implement the interface. The class has an instance variable called `balance`. Define the method to compute interest. The interest rate for saving account is 3% per year. Create a `Tester` class to test the program.
3. Create a class `Person` that implements the `comparable` interface. The class has an instance variable `name`. The class has the constructor that initializes the name. The class also has the accessor method and a display method to display the name. Create an array for multiple `Person` objects. Sort the person in ascending order. Create a `Tester` class to test the program.